



ELSEVIER

Journal of Computational and Applied Mathematics 53 (1994) 87–108

JOURNAL OF
COMPUTATIONAL AND
APPLIED MATHEMATICS

Some outer approximation methods for semi-infinite optimization problems

Rembert Reemtsen *

Fachbereich Mathematik (MA 6-3), Technische Universität Berlin, Straße des 17. Juni 136, D-10623 Berlin, Germany

Received 9 September 1992; revised 1 October 1992

Abstract

The paper starts with a simple model and convergence theorem for outer approximation methods. This general framework is used to unify and derive and modify certain exchange methods, cutting methods and discretization methods for semi-infinite programming problems. By that, in particular, a cutting plane method for convex semi-infinite programs is developed. For a practically reasonable specification (the method is more generally stated), the subproblems in the given algorithm are moderately sized quadratic problems, and each step of the algorithm can be performed by means of finitely many operations.

Keywords: Semi-infinite programming; Outer approximation methods; Exchange methods; Cutting methods; Discretization methods; Convex programming

0. Introduction

We begin with stating the problem which we consider here. We let $A \subseteq \mathbb{R}^n$ be a nonempty closed set of parameters and $f: A \rightarrow \mathbb{R}$ and $g: A \rightarrow C(B)$ be continuous mappings where B is a compact set of \mathbb{R}^s and $C(B)$ is equipped with the usual supremum norm; the spaces \mathbb{R}^n and \mathbb{R}^s are associated with an l_p -norm $\|\cdot\|_p$, $1 \leq p \leq \infty$, or an arbitrary norm $\|\cdot\|$, respectively. Then, for each set $D \subseteq B$ and for each $\delta \geq 0$, we define a feasible set

$$Z_\delta(D) := \{a \in A \mid g(a, x) \leq \delta, x \in D\},$$

and we assign in particular to $Z_0(D)$ the optimization problem

$$S[D]: \quad \mu(D) := \inf_{a \in Z_0(D)} f(a).$$

* e-mail: reemtsen@math.tu-berlin.de.

We call $S[D]$ linear when $A := \mathbb{R}^n$ and f and g are affine linear, and we say that $S[D]$ is convex if A is convex and f and g are convex mappings. The results of this paper can in an obvious way be extended to problems in which the feasible set is formed by the set A and finitely many mappings $g_j: A \rightarrow C(B^j)$ with $B^j \subseteq \mathbb{R}^{s_j}$ being a compact set. (See also [31] in this respect.)

If D has a finite cardinality $|D|$, then $S[D]$ is a finite optimization problem. Conversely, each ordinary finite programming problem can be written in the form of $S[D]$. In this paper we are concerned with the solution of the problem $S[B]$ where we normally have the semi-infinite programming problem in mind, i.e., the case where $|B| = \infty$. However, the infinity of $|B|$ is never used here, so that our results also apply to finite optimization problems.

The purpose of this paper is to unifyingly treat some closely related methods for the solution of semi-infinite programming problems and to develop, in particular, a cutting plane method for convex semi-infinite programs. The methods considered here are Remez-type algorithms (also called exchange methods, e.g., in [14,17] or semi-continuous type methods in [21]), certain discretization methods (solving $S[D_k]$, $k = 0, 1, \dots$, where D_k is a subset of a grid $B_k \subseteq B$) and cutting methods. (Relations between them have, for example, been pointed out in [3,14,17,32].) These methods have in common that they solve the semi-infinite program $S[B]$ by replacing it by a sequence of finite minimization problems (P_k) , $k = 0, 1, \dots$, where the feasible set of (P_k) is formed by g and an appropriate finite set $D_k \subseteq B$ and encloses the set of parameters $Z_0(B)$ of $S[B]$. (For linear problems, (P_k) typically coincides with $S[D_k]$ and so has the feasible set $Z_0(D_k) \supseteq Z_0(B)$.) Consequently, we can regard each of these methods as a special outer approximation method for $S[B]$, which is the point of view that we want to take here. (Outer approximation methods for semi-infinite programming problems, which are of different type than those studied here, can be found, e.g., in [9].)

Outer approximation methods for optimization problems have been studied in quite general settings (see [5,18–20,23,27,29,37–39]) where the frame in [5,18] seems to be the most general (see [19] in this respect). However, instead of dealing with set-valued mappings and cut maps as used in [5,18], we choose here to provide a simple general convergence theorem for outer approximation methods (in Section 1) by considering their convergence as a special problem of discrete approximations (cf. [11]) and to apply it afterwards to the algorithms in which we are interested here. Our approach reveals that the objective function itself may be perturbed in an outer approximation method (which has been utilized in [7,8] in a specific way) and that neither monotonicity of the minimal values nor inclusion of the feasible sets of the involved subproblems is needed in order to prove its convergence. (The latter has, for example, been assumed in [23,27].) We employ these facts here especially in the announced algorithm for general convex semi-infinite programming problems. Approximation of the objective function may also help to simplify or to stabilize the finite subproblems of a method which actually have to be solved numerically.

Outer approximation methods usually require the knowledge of a compact set which encloses the feasible set of the problem being investigated. In Section 2 we show firstly how the feasible set $Z_0(B)$ of the semi-infinite programming problem $S[B]$ is equivalently replaced by a set which we assume to be bounded (and which in the convex case normally is bounded), and secondly how a compact envelopment of this latter set can be constructed.

In Section 3 we consider again the Remez-type discretization method for general nonlinear

semi-infinite programming problems from [32]. As it turns out, it can be interpreted as an outer approximation method which employs nonlinear cuts and in which a compact enclosure of the feasible set is used implicitly. An obvious modification of this algorithm is offered for the case where g is a convex mapping.

The number of constraints in the algorithms of Section 3 increases with each iteration. It is well known that for a finite convex optimization problem the nonbinding constraints at each iteration of an outer approximation method can be dropped if its objective function is strictly convex. (See, e.g., [5,7–9,19,37,38] on constraint dropping techniques.) In Section 4 we modify the algorithms of Section 3 for semi-infinite programming problems with an “almost strictly convex” goal function in this sense.

In [32] we have given a discretization method for linear semi-infinite optimization problems which has been shown to be capable of solving quite large problems of this type. The main intention of this paper is to prove the convergence of a similar method for general convex semi-infinite programs. This is finally done in Section 5 where we make use of the results obtained before. For a specification of the given algorithm (it is more generally stated), the subproblems in the algorithm are reasonably sized finite quadratic problems and each step in the algorithm can be numerically carried out by means of finitely many operations. An application of the algorithm in Section 5 to the solution of complex minimax problems can be found in [30,33].

Our approach here enables us in particular to discuss a number of previously published methods in a unifying context and to modify them in one or more ways explained next. At first, the algorithms here offer the option of approximating the objective function of $S[B]$ in a certain manner, as we mentioned already, and a way of constructing a compact set, which encloses the feasible region of interest, is described explicitly. Besides, the occurring finite optimization subproblems have only to be solved approximately. Two further points are outlined in the following.

A particular difficulty in the numerical solution of semi-infinite programming problems is to determine whether an element $a \in A$ is a feasible point for $S[B]$, or to compute the (possibly most violating) constraint function $\max\{g(a, x) \mid x \in B\}$, respectively, as is required at each iteration by most algorithms. Up to now there does not exist a numerical method which can solve this global optimization problem in the general case (cf. [20]), and it is, therefore, suggested in the literature to first search for all local extrema of $g(a, \cdot)$ with respect to B on a sufficiently dense grid and to compute each of them afterwards approximately, e.g., by Newton’s method. In order to obtain algorithms, in which each step can be executed through a finite number of operations, we alternatively select here (as in [32]) a sequence B_k , $k \in \mathbb{N}$, of finite sets converging to B , and we evaluate g at the k th iteration only on the grid B_k . (The B_k ’s may be arbitrary compact subsets of B so that $B_k := B$, $k \in \mathbb{N}$, like in most algorithms, becomes one possible choice.)

The computation of the global maximum of g on B or a grid $B_k \subseteq B$ at each iteration is the numerically most costly step in the methods discussed here, at least for multi-dimensional regions B . Consequently the number of iterations being needed by such an algorithm, in order to provide a solution with a requested accuracy, should be kept as small as possible. So algorithms of the type considered here which employ only one new constraint at each iteration turn out to be extremely inefficient. (This has, for example, been reported in [4] for the related KCG cutting plane method of Kelley [24], Chency and Goldstein [3]. With respect to semi-in-

finite programming problems see also the remarks in [14, p.163].) Without additional effort, the algorithms here can be stated such that they grant the option of adding more constraints at each iteration if this is requested.

The methods being studied in this paper require the computation of that constraint at each iteration which is most violated in a certain sense, and they are, therefore, related to the KCG-algorithm. With the general approach used here, one can easily modify some of these algorithms by employing cuts as they are, for example, performed in the supporting hyperplane method of [40]. However, there is no indication that such modifications will in general be numerically more efficient. We refer in this connection again to the investigations in [4]. (Note that for the numerical examples in [32] not only all violated constraints were used at each iteration, as is recommended by [4], but also certain nonviolated ones.) Similar remarks can be made for the cutting plane methods in [10,35] which are based on the central cutting plane algorithm in [6]. (These methods do not represent outer approximation methods for $S[B]$ directly, but for the related $(n+1)$ -dimensional problem inherent in those algorithms.)

Outer approximations of a solution are naturally obtained in semi-infinite programming whenever only a finite number of the infinitely many constraints is taken into consideration. The methods being studied here converge globally and under mild assumptions, and in concrete situations they, therefore, may sometimes be the only methods which are applicable at all. (For other recent techniques in semi-infinite optimization we merely refer here to [28,34] and the references given there.) Unfortunately, outer approximation methods of the considered type can become numerically instable in a neighborhood of the solution (cf., e.g., [17,32]), in which case they serve at least to determine a starting point for a locally convergent and usually faster method (see [14,17]). The option of perturbing the underlying objective function, which is offered here, may also disclose a chance to overcome this numerical difficulty.

1. A general model for outer approximation methods

We start from an optimization problem of the form

$$(P): \quad \rho := \inf_{a \in M} f(a),$$

where

$$M \neq \emptyset, \quad M \subseteq \mathbb{R}^n \text{ is compact}, \quad (1.1)$$

and f is a continuous functional containing M in its domain of definition so that the infimum in (P) is achieved for some $a \in M$. In this section we provide a general convergence theorem for a broad class of outer approximation methods for the solution of (P), which will be the basic tool for the convergence proofs of the algorithms in the subsequent sections.

In outer approximation methods, (P) is typically replaced by a sequence of (normally simpler) problems

$$(P_k): \quad \rho_k := \inf_{a \in M_k} f_k(a), \quad k \in \mathbb{N}_0,$$

where $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$ and M_k and f_k are suitable approximations of M and f , respectively. In particular, M_k is constructed in such a way that

$$M \subseteq M_k \subseteq X, \quad k \in \mathbb{N}_0, \text{ with compact } X \subseteq \mathbb{R}^n, \quad (1.2)$$

where the set X usually needs to be known explicitly and often is assumed to be polyhedral because of numerical reasons. In classical methods, the function f_k normally is chosen equal to f for all $k \in \mathbb{N}_0$. If for each $k \in \mathbb{N}_0$ the set M_k is obtained by intersecting the compact envelopment X of M with the solution set of finitely many inequality constraints, such procedures are also referred to as cutting methods.

In practice, one can often compute only an approximate solution of (P_k) which is element of a set $\tilde{M}_k \subseteq \tilde{X}$ where \tilde{M}_k and \tilde{X} are suitable approximations of M_k and X with

$$\tilde{M}_k \subseteq \tilde{X}, \quad k \in \mathbb{N}_0, \text{ for compact } \tilde{X}, \quad X \subseteq \tilde{X} \subseteq \mathbb{R}^n. \quad (1.3)$$

Thus with (1.2) we have the inclusions

$$M \subseteq M_k \subseteq X \subseteq \tilde{X}, \quad \tilde{M}_k \subseteq \tilde{X},$$

where usually $M_k \subseteq \tilde{M}_k$, $k \in \mathbb{N}_0$, for example, when it is not possible to exactly satisfy constraints describing M_k . We finally assume now that

$$f: \tilde{X} \rightarrow \mathbb{R} \text{ is continuous, } f_k: \tilde{X} \rightarrow \mathbb{R}, \quad k \in \mathbb{N}_0. \quad (1.4)$$

So if, moreover, M_k is closed (which is the case in the following sections) and likewise f_k is continuous on \tilde{X} , the problem (P_k) also possesses a solution. (In this paper quantities with a tilde represent approximations of the quantities denoted by the same symbol without tilde and hence may be set identical with these for a first simpler understanding of the results. The set X is not needed here further. We use two compact sets X and \tilde{X} in the following sections for reasons of symmetry and for being able to express that also constraints describing X have not to be satisfied exactly.)

In order to next derive results on the convergence of (approximate) solutions of (P_k) , we first give some definitions which simplify the presentation. For every subset $H \subseteq \mathbb{N}_0$, $H = \{k_0, k_1, k_2, \dots\}$ with $k_i \leq k_{i+1}$, $i \in \mathbb{N}_0$, $k_i \rightarrow \infty$, $i \rightarrow \infty$, and for $a \in \mathbb{R}^n$, $a_k \in \mathbb{R}^n$, $k \in \mathbb{N}_0$, we write

$$a_k \rightarrow a, \quad k \in H : \Leftrightarrow a_{k_i} \rightarrow a, \quad i \rightarrow \infty.$$

We let further \mathbb{N}' , \mathbb{N}'' etc. be naturally ordered subsets of \mathbb{N}_0 , not necessarily meaning the same sets at different occurrences. For mappings $g: S \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$, $g_k: S_k \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$, $k \in \mathbb{N}_0$, we use the notation $g_k|S_k \rightarrow g|S$, $k \in \mathbb{N}_0$, if

$$\mathbb{N}' \subseteq \mathbb{N}_0, \quad a_k \in S_k, \quad a_k \rightarrow a \in S, \quad k \in \mathbb{N}' \Rightarrow g_k(a_k) \rightarrow g(a), \quad k \in \mathbb{N}'.$$

Definition 1.1. Given (1.1) and (1.4) with $M \subseteq \tilde{X} \subseteq \mathbb{R}^n$, a sequence $\tilde{a}_k \in \tilde{X}$, $k \in \mathbb{N}_0$, is said to be a *minimizing sequence* for (P) if

- (i) $f_k(\tilde{a}_k) \rightarrow \rho$, $k \in \mathbb{N}_0$,
- (ii) $\mathbb{N}' \subseteq \mathbb{N}_0$, $\tilde{a}_k \rightarrow a \in \tilde{X}$, $k \in \mathbb{N}' \Rightarrow a \in M$, $f(a) = \rho$.

A sequence $\tilde{a}_k \in \tilde{X}$, $k \in \mathbb{N}_0$, can only be minimizing for (P) if at least $f_k | \{\tilde{a}_k\} \rightarrow f | \tilde{X}$, $k \in \mathbb{N}_0$, is given. Clearly, if \tilde{X} is compact, a minimizing sequence $\tilde{a}_k \in \tilde{X}$, $k \in \mathbb{N}_0$, possesses an accumulation point, and if further (P) has a unique solution $\hat{a} \in M$, then $\tilde{a}_k \rightarrow \hat{a}$, $k \in \mathbb{N}_0$, holds true. With these preliminaries we can now obtain the following convergence theorem.

Theorem 1.2. *Let (1.1)–(1.4) be satisfied and let*

$$f_k | \tilde{X} \rightarrow f | \tilde{X}, \quad k \in \mathbb{N}_0. \quad (1.5)$$

If $\tilde{a}_k \in \tilde{M}_k$, $k \in \mathbb{N}_0$, is a sequence such that

$$\mathbb{N}' \subseteq \mathbb{N}_0, \quad \tilde{a}_k \rightarrow a \in \tilde{X}, \quad k \in \mathbb{N}' \Rightarrow a \in M \quad (1.6)$$

and

$$(f_k(\tilde{a}_k) - \rho_k) \rightarrow 0, \quad k \in \mathbb{N}_0, \quad (1.7)$$

then \tilde{a}_k , $k \in \mathbb{N}_0$, is a minimizing sequence for (P).

Proof.¹ Let $a^* \in M$ be an optimal element for (P). Then, due to (1.2), a^* is in M_k so that

$$\rho_k \leq f(a^*) + (f_k(a^*) - f(a^*)),$$

which by (1.2)–(1.5) implies

$$\limsup_{k \in \mathbb{N}_0} \rho_k \leq \rho. \quad (1.8)$$

Utilizing (1.7), we arrive at

$$\limsup_{k \in \mathbb{N}_0} f_k(\tilde{a}_k) \leq \rho.$$

Now let $\mathbb{N}' \subseteq \mathbb{N}_0$ and $a \in \tilde{X}$ be such that $\tilde{a}_k \rightarrow a$, $k \in \mathbb{N}'$. Then with (1.6) and (1.8) we obtain

$$\rho \leq f(a) = \lim_{k \in \mathbb{N}'} f_k(\tilde{a}_k) \leq \limsup_{k \in \mathbb{N}_0} f_k(\tilde{a}_k) \leq \rho. \quad (1.9)$$

Thus $f(a) = \rho$ and $f_k(\tilde{a}_k) \rightarrow \rho$, $k \in \mathbb{N}_0$, since (1.9) holds true for every converging subsequence of \tilde{a}_k , $k \in \mathbb{N}_0$. \square

Remark 1.3. Under the assumptions of Theorem 1.2 the convergence (1.5) is equivalent to the uniform convergence

$$\sup_{a \in \tilde{X}} |f_k(a) - f(a)| \rightarrow 0, \quad k \in \mathbb{N}_0.$$

¹ In the first version of this paper, Theorem 1.2 had been proved by a straightforward application of the stability theory in [11]. For convenience, we give here a nice short direct proof instead, which was suggested to us by Prof. Dr. A. Kirsch of the University of Erlangen-Nürnberg.

If f and f_k , $k \in \mathbb{N}_0$, are convex functions on \mathbb{R}^n , a sufficient condition for (1.5) is given by the pointwise convergence

$$f_k(a) \rightarrow f(a), \quad k \in \mathbb{N}_0, \quad a \in \mathbb{R}^n,$$

(see [25]). Finally, in case $f_k(a) \leq f(a)$, $a \in \tilde{X}$, holds true for all $k \in \mathbb{N}_0$ (which implies (1.8) by (1.2)), it is easily seen that instead of (1.5) it suffices to assume

$$f_k|_{\{\tilde{a}_k\}} \rightarrow f|_{\tilde{X}}, \quad k \in \mathbb{N}_0,$$

for the sequence in Theorem 1.2 satisfying (1.7).

If a minimizing sequence for (P) having property (1.7) exists, one can also infer the convergence $\rho_k \rightarrow \rho$, $k \in \mathbb{N}_0$. For $M_k \subseteq \tilde{M}_k$, $k \in \mathbb{N}_0$, as in the following sections, condition (1.7) is in particular satisfied if \tilde{a}_k , $k \in \mathbb{N}_0$, is in M_k and solves (P_k) . We finally mention that estimates for $|\rho - \rho_k|$ can be found in [11].

2. The construction of an enclosing compact set

Outer approximation methods for the solution of optimization problems, as they are treated in Section 1, are based on the knowledge of a compact set X enclosing the respective set of feasible points. In this section we show how such a set possibly can be determined for the semi-infinite programming problem $S[B]$.

Normally the feasible set $Z_0(B)$ of $S[B]$ is unbounded. So we first cut off a part of $Z_0(B)$ which does not contain any solution of $S[B]$. This is achieved by defining for each $a_0 \in Z_0(B)$ and $D \subseteq B$ the level set

$$L(a_0, D) := \{a \in A \mid f(a) \leq f(a_0)\} \cap Z_0(D), \quad (2.1)$$

for which obviously

$$\mu(D) = \inf_{a \in Z_0(D)} f(a) = \inf_{a \in L(a_0, D)} f(a). \quad (2.2)$$

Thus we can replace the feasible set $Z_0(B)$ in $S[B]$ equivalently by some set $L(a_0, B)$ which we assume now to be bounded.

(A0) For some $a_0 \in Z_0(B)$, the set $L(a_0, B)$ is bounded (and hence compact). Further, compact sets X and \tilde{X} are given such that

$$L(a_0, B) \subseteq X \subseteq \tilde{X} \subseteq A. \quad (2.3)$$

By (2.2), assumption (A0) guarantees the existence of a solution of $S[B]$. For convex programs (A0) usually is satisfied, but (A0) may fail to hold in the general nonlinear situation. In case the set $L(a_0, B)$ is unbounded for each $a_0 \in Z_0(B)$, a compact feasible set can still be found if a priori information on a solution of the problem is available (cf. [32]).

The sets X and \tilde{X} in (A0) will later on play the same role as X and \tilde{X} in Section 1. For practical reasons, they will normally have to be polyhedral sets, where for $X := \{a \in \mathbb{R}^n \mid a^T b_j +$

$d_j \leq 0, j \in I\}$ with $b_j \in \mathbb{R}^n, d_j \in \mathbb{R}, j \in I, I := \{1, \dots, m\}$, the set \tilde{X} may be defined as $\tilde{X} := \{a \in \mathbb{R}^n \mid a^T b_j + d_j \leq \epsilon, j \in I\}$ for some $\epsilon \geq 0$. (It is easily seen that if X is a polyhedron, so is \tilde{X} .) Therefore, in order to explicitly construct such sets, it will usually be helpful to start from an upper set of $L(a_0, B)$, which can be described by finitely many constraints (under the assumption that A has this property). For that we let $f_k, k \in \mathbb{N}_0$, be functions on A ,

$$h_1(a) := \min\left(f(a), \inf_{k \in \mathbb{N}_0} f_k(a)\right), \quad a \in A, \quad (2.4a)$$

$$h_2(a) := \max\left(f(a), \sup_{k \in \mathbb{N}_0} f_k(a)\right), \quad a \in A, \quad (2.4b)$$

and, in analogy to (2.1),

$$\Lambda(a_0, D) := \{a \in A \mid h_1(a) \leq h_2(a_0)\} \cap Z_0(D),$$

for $a_0 \in Z_0(B)$ and $D \subseteq B$. We further assume (in regard to the subsequent sections) that a sequence of (usually finite) sets $B_k, k \in \mathbb{N}_0$, is given where

$$\begin{aligned} B_k \subseteq B \text{ is compact, } B_k \subseteq B_{k+1}, \quad k \in \mathbb{N}_0, \\ |B_0| < \infty, \quad \sup_{x \in B} \inf_{y \in B_k} \|x - y\|_\infty \rightarrow 0, \quad k \in \mathbb{N}_0, \end{aligned} \quad (2.5)$$

and we state the following assumption.

(A1) $B_0 \subseteq B$ in (2.5) is such that, for some $a_0 \in Z_0(B)$, the set $\Lambda(a_0, B_0)$ is compact. Furthermore, compact sets X and \tilde{X} are given with

$$\Lambda(a_0, B_0) \subseteq X \subseteq \tilde{X} \subseteq A. \quad (2.6)$$

It is easily seen that for each D with $B_0 \subseteq D \subseteq B$ we have

$$L(a_0, D) \subseteq \Lambda(a_0, D), \quad \Lambda(a_0, B) \subseteq \Lambda(a_0, D) \subseteq \Lambda(a_0, B_0), \quad (2.7)$$

and that $\Lambda(a_0, D) = L(a_0, D)$ if $f_k := f, k \in \mathbb{N}_0$. Eq. (2.7) in particular shows that (A1) implies (A0). With

$$L_k := \{a \in A \mid f_k(a) \leq f_k(a_0)\}$$

we, moreover, have the inclusion $(L_k \cap Z_0(D)) \subseteq \Lambda(a_0, D)$ and

$$\inf_{a \in Z_0(D)} f_k(a) = \inf_{a \in (L_k \cap Z_0(D))} f_k(a) = \inf_{a \in \Lambda(a_0, D)} f_k(a). \quad (2.8)$$

Thus, for f_k being continuous on A , assumption (A1) further guarantees by (2.7) that $L_k \cap Z_0(D)$ is compact and consequently the infimum in (2.8) is achieved for some $a \in Z_0(D)$. We note that for linear $S[B]$ assumption (A1) implies $|B_0| \geq n$. If merely (A0) is provided, the set B_0 in (2.5) can be chosen arbitrarily and may even be equal to the empty set.

We conclude with some remarks concerning the study of convex problems in Sections 4 and 5. So we assume $g, f, f_k, k \in \mathbb{N}_0$, to be continuous mappings on convex A , where A is required here to be determined by finitely many convex inequality constraints. Then, if h_1 is a convex function, $\Lambda(a_0, B_0)$ is the solution set of finitely many convex inequality constraints

where for practical purposes h_1, h_2 (2.4) need to be given by simple instructions. (This is, for instance, the case under (A3) or for the choices (5.8), (5.9) under (A4) below.) A suitable polyhedron X then can usually be constructed by proper linearizations of the functions defining $\Lambda(a_0, B_0)$. A polyhedral set, tightly enveloping $\Lambda(a_0, B_0)$, may also be derived from the solution of certain $n + 1$ convex programming problems as is shown in [20, p.69]. Finally, if f and g are linear, $A := \mathbb{R}^n$ and $h_1 = f$ (as for (5.8)), one can choose $X := \Lambda(a_0, B_0)$.

3. Methods for nonlinear problems

We begin with modifying the algorithm in [32] for the solution of the general nonlinear semi-infinite programming problem $S[B]$ defined in Section 0. For the sake of simplicity of the presentation, the algorithms here are formulated without a stopping criterion and hence generate an infinite sequence of iterates. A stopping criterion for the following algorithm will be discussed afterwards in Remark 3.3.

Algorithm I. Assume (A0) to be satisfied. For $k \in \mathbb{N}_0$, let the sets B_k be as in (2.5), let $f_k : \tilde{X} \rightarrow \mathbb{R}$ be arbitrary continuous functions fulfilling (1.5) and let ε_k, δ_k be such that

$$\min(\varepsilon_k, \delta_k) \geq 0, \quad k \in \mathbb{N}_0, \quad \max(\varepsilon_k, \delta_k) \rightarrow 0, \quad k \in \mathbb{N}_0. \quad (3.1)$$

Step 0. Set $M_0 := X, \tilde{M}_0 := \tilde{X}, D_0 := B_0$, and $k := 0$.

Step 1. For

$$(P_k): \quad \rho_k := \inf_{a \in M_k} f_k(a), \quad (3.2)$$

find an (approximate) solution such that $\tilde{a}_k \in \tilde{M}_k$ and

$$|f_k(\tilde{a}_k) - \rho_k| \leq \varepsilon_k. \quad (3.3)$$

Step 2. Select $D_{k+1} \subseteq B_{k+1}$ with $D_{k+1} \supseteq D_k \cup \{x_k\}$ where $x_k \in B_{k+1}$ satisfies

$$g(\tilde{a}_k, x_k) = \max_{x \in B_{k+1}} g(\tilde{a}_k, x), \quad (3.4)$$

and, defining

$$N_k(\delta) := Z_\delta(D_{k+1}) = \{a \in A \mid g(a, x) \leq \delta, x \in D_{k+1}\}, \quad (3.5)$$

for $\delta \geq 0$, set

$$M_{k+1} := X \cap N_k(0), \quad \tilde{M}_{k+1} := \tilde{X} \cap N_k(\delta_{k+1}). \quad (3.6)$$

Step 3. Set $k := k + 1$ and go to Step 1.

Theorem 3.1. Algorithm I generates a minimizing sequence $\tilde{a}_k \in \tilde{X}$, $k \in \mathbb{N}_0$, for $S[B]$. In particular, for $k \in \mathbb{N}_0$, we have

$$\rho_k \leq \rho_{k+1} \leq \mu(B), \quad \text{if } f_k(a) \leq f_{k+1}(a) \leq f(a), \quad a \in \tilde{X}. \quad (3.7)$$

Proof. Due to (2.2) we can choose $M := L(a_0, B)$ as the feasible set of $S[B]$ which by virtue of (A0) is compact and satisfies $M \subseteq X \subseteq \tilde{X} \subseteq A$. Then, since $B_0 = D_0 \subseteq D_k \subseteq B$, $k \in \mathbb{N}_0$, for the choice (3.6) obviously conditions (1.2) and (1.3) are fulfilled. Because of $\{a_0\} \subseteq M \subseteq M_k \subseteq \tilde{M}_k$ (a_0 as in (A0)), there further exists an element $\tilde{a}_k \in \tilde{M}_k$ with (3.3). Hence, in order to be able to employ Theorem 1.2, we still need to verify (1.6), i.e.,

$$\mathbb{N}' \subseteq \mathbb{N}_0, \tilde{a}_k \rightarrow \tilde{a} \in \tilde{X}, k \in \mathbb{N}' \Rightarrow \tilde{a} \in L(a_0, B). \quad (3.8)$$

By (2.3), \tilde{a} lies in A , and by (3.3),

$$f_k(\tilde{a}_k) - \varepsilon_k \leq \rho_k \leq f_k(a_0), \quad k \in \mathbb{N}_0,$$

holds true, which implies $f(\tilde{a}) \leq f(a_0)$ because of (1.5) and (3.1). Next, the convergence of B_{k+1} , $k \in \mathbb{N}'$, implies that for each $\xi \in B$ we can find $\xi_k \in B_{k+1}$, $k \in \mathbb{N}'$, with $\xi_k \rightarrow \xi$, $k \in \mathbb{N}'$. There further exist $\mathbb{N}'' \subseteq \mathbb{N}'$ and $\tilde{x} \in B$ such that

$$x_k \rightarrow \tilde{x}, \quad k \in \mathbb{N}''. \quad (3.9)$$

Therefore, from $g(\tilde{a}_k, x_k) \geq g(\tilde{a}_k, \xi_k)$, $k \in \mathbb{N}''$, we obtain $g(\tilde{a}, \tilde{x}) \geq g(\tilde{a}, \xi)$.

Now for $l \in \mathbb{N}''$, $l > k$, we have $x_k \in D_l$ and consequently

$$g(\tilde{a}_l, x_k) \leq \delta_l,$$

which finally yields

$$g(\tilde{a}, \tilde{x}) \leq 0. \quad (3.10)$$

At last, (3.7) is a consequence of the inclusions $M \subseteq M_{k+1} \subseteq M_k$, $k \in \mathbb{N}_0$. \square

Remark 3.2. If instead of (A0) assumption (A1) is given, the sets M_k , $k \in \mathbb{N}$, in the algorithm can as well be replaced by $M_k := Z_0(D_k)$, $k \in \mathbb{N}$, so that for $f_k := f$, $k \in \mathbb{N}$, the problem (P_k) (3.2) just is the problem $S[D_k]$; for, observe that by (2.8) for all $k \in \mathbb{N}$ we have

$$\rho_k = \inf_{a \in (X \cap Z_0(D_k))} f_k(a) = \inf_{a \in \Lambda(a_0, D_k)} f_k(a) = \inf_{a \in Z_0(D_k)} f_k(a). \quad (3.11)$$

If we, moreover, set $X := \Lambda(a_0, B_0)$, this is also true for $k = 0$. In that case, Algorithm I essentially becomes [32, Algorithm 2.1]. If further for a $\delta \geq \max(\varepsilon_k, \delta_k)$, $k \in \mathbb{N}_0$, and for h_1, h_2 (2.4) the set

$$\Delta := \{a \in A \mid h_1(a) \leq h_2(a_0) + \delta\} \cap Z_\delta(B_0),$$

enclosing $\Lambda(a_0, B_0)$, is compact, we may analogously choose $\tilde{M}_k := Z_{\delta_k}(D_k)$. (Let $\tilde{X} := \Delta$ in (A1) and note that then each $\tilde{a}_k \in Z_{\delta_k}(D_k)$ with (3.3) lies in \tilde{X} .) We note that for the proof of Theorem 3.1 (needed in Section 5) we could not employ certain monotonicity arguments as they have been used in the convergence proof of [32, Algorithm 2.1] and also by other authors for this type of algorithms.

Remark 3.3. Instead of (3.4), it suffices to find $x_k \in B_{k+1}$ with

$$g(\tilde{a}_k, x_k) \geq \max_{x \in B_k} g(\tilde{a}_k, x).$$

Further, a numerically reasonable choice of the sets $B_k, D_k, k \in \mathbb{N}_0$, can be obtained by employing the ideas of [32, Section 3]. (See also Remark 4.5 below.) From the proof of Theorem 3.1 it is clear that an appropriate stopping criterion for Algorithm I is that for some $\vartheta > 0$ and $k_0 \in \mathbb{N}$ the conditions

$$g(\tilde{a}_k, x_k) \leq \vartheta \quad \text{and} \quad k \geq k_0 \quad (3.12)$$

are simultaneously satisfied. (For $k \geq k_0$, one may set $B_{k+1} = B_k$ until the first inequality in (3.12) is fulfilled. In this way an approximate solution of $S[B_{k_0}]$ is obtained.)

Remark 3.4. As Remark 3.2 shows, for $B_k := B, k \in \mathbb{N}$, and $f_k := f, k \in \mathbb{N}_0$, Algorithm I basically is the nonlinear semi-infinite programming version of the first Remez algorithm which in the thirties was proposed by Remez for the solution of the linear Chebyshev approximation problem (see [3,31], also for further references concerning this particular problem). With special choices of the parameters (almost always $B_k := B, f_k := f, f(\tilde{a}_k) := \rho_k, D_{k+1} := D_k \cup \{x_k\}, k \in \mathbb{N}$, and $\tilde{X} := X$ being unspecified if used), it has been suggested for linear semi-infinite programming problems in [3,12–14,21] and for nonlinear problems in [1,2,21]. The version of Algorithm I here is a modification of [32, Algorithm 2.1].

Remark 3.5. If, for example, all functions $f_k, k \in \mathbb{N}_0$, are Lipschitz continuous on \tilde{X} with a uniform Lipschitz constant C (and so by (1.5) is f), then

$$|f_k(\tilde{a}_k) - \mu(B)| \leq \varepsilon_k + \max_{a \in M} |f(a) - f_k(a)| + C \max_{a \in M_k} \min_{b \in M} \|a - b\|,$$

with $M = L(a_0, B)$ and $M_k = X \cap Z_0(D_k)$ or, by (3.11), $M_k = \Lambda(a_0, D_k)$ respectively if (A1) is satisfied. (Use (3.3), the inclusion $M \subseteq M_k$, and [11, (7)].)

In case g is a convex function, instead of nonlinear cuts as in Algorithm I also linear cuts may be used. So in addition to our assumptions of Section 0 we assume the following.

(A2) A is convex, and g is convex on an open upper set A_0 of A .

Under assumption (A2), there exists, for each fixed $x \in B$, a “subgradient” $\partial g(\hat{a}, x) \in \mathbb{R}^n$ for $\hat{a} \in A$ satisfying

$$g(a, x) - g(\hat{a}, x) \geq \partial g(\hat{a}, x)^T (a - \hat{a}), \quad a \in A_0, \quad (3.13)$$

(cf., e.g., [2]). It is easily seen that the set of subgradients $\{\partial g(a, x) \in \mathbb{R}^n \mid a \in E, x \in B\}$ is compact for each compact $E \subseteq A$. We now are in the position to prove convergence of the following algorithm where (as also later on) $\partial g(a, x)$ is an arbitrary subgradient for $a \in A$ given.

Algorithm II. As Algorithm I with (3.5) exchanged for

$$N_k(\delta) := \left\{ a \in A \mid g(\tilde{a}_j, x) + \partial g(\tilde{a}_j, x)^T (a - \tilde{a}_j) \leq \delta, \forall x \in D_{j+1} \setminus D_j, j = 0, \dots, k \right\}, \quad (3.14)$$

where in addition (A2) is assumed.

Corollary 3.6. *The sequence $\tilde{a}_k \in \tilde{X}$, $k \in \mathbb{N}_0$, induced by Algorithm II, is a minimizing sequence for $S[B]$. Furthermore, for ρ_k , $k \in \mathbb{N}_0$, the relationship (3.7) is valid.*

Proof. We can follow the proof of Theorem 3.1 up to formula (3.9) where $M \subseteq M_k$ here is a consequence of inequality (3.13). Next we observe that

$$g(\tilde{a}_k, x_k) + \partial g(\tilde{a}_k, x_k)^T(\tilde{a}_l - \tilde{a}_k) \leq \delta_l, \quad l > k, \quad k, l \in \mathbb{N}'',$$

and that, for some $K > 0$, consequently with (3.1)

$$g(\tilde{a}_k, x_k) \leq K \|\tilde{a} - \tilde{a}_k\|, \quad k \in \mathbb{N}'', \quad (3.15)$$

holds true. By employing (3.9) and by taking the limit in (3.15), we arrive at (3.10). \square

Remark 3.7. Remark 3.3 remains valid for Algorithm II. We further note that any constraint in $N_k(\delta)$ (3.14) with $\partial g(\tilde{a}_j, x) = 0$ can be cancelled. (Since $Z_0(B) \neq \emptyset$, this case can only occur if $g(\tilde{a}_j, x) \leq 0$.) For $f_k := f$, $B_k := B$, $k \in \mathbb{N}_0$, and B being a finite set, Algorithm II essentially is the cutting plane method of Kelley [24], Cheney and Goldstein [3].

4. Methods for convex programs with an almost strictly convex objective function

The algorithms of the previous section converge for general nonlinear semi-infinite programming problems or for nonlinear problems with a convex constraint mapping g , respectively. Due to the increase of the number of constraints in the finite subproblems appearing in the algorithms, the practical applicability of these methods, however, is restricted to relatively coarse grids, i.e., normally to a relatively small $k_0 \in \mathbb{N}$ in the stopping criterion (3.12). On the other hand, it has been shown in [5,37,38] for cutting methods solving finite convex programs that the nonbinding, i.e., the inactive constraints can be dropped in each iteration if the objective function fulfils some strict convexity assumption. In this section we pursue this idea in order to develop algorithms for the solution of related convex semi-infinite programming problems, where we make use again of the general framework provided in Section 1.

The main reason why dropping of inactive constraints is permitted for finite convex programs is explained by the following lemma. (Note that a convex function is strictly quasi-convex.)

Lemma 4.1. *Let $c_0: A \rightarrow \mathbb{R}$ be strictly quasi-convex and $c_j: A \rightarrow \mathbb{R}$, $j \in J := \{1, \dots, n\}$, be convex on the convex set A . Further, let*

$$F(I) := \{a \in A \mid c_j(a) \leq 0, j \in I\}, \quad I \subseteq J, \quad I(a) := \{j \in J \mid c_j(a) = 0\}, \quad a \in A.$$

If $\hat{a} \in F(J)$ with $c_0(\hat{a}) = \inf\{c_0(a) \mid a \in F(J)\}$ exists, then

$$c_0(\hat{a}) = \inf_{a \in F(J)} c_0(a) = \inf_{a \in F(I(\hat{a}))} c_0(a).$$

Proof. For $I(\hat{a}) = \emptyset$, i.e., $F(I(\hat{a})) = A$, the proposition follows immediately from the strict quasi-convexity of c_0 . In case $I(\hat{a}) \neq \emptyset$, one can easily conclude from the convexity of A and c_j , $j \in J$, that for each $\tilde{a} \in F(I(\hat{a}))$ there is a $\lambda \in (0, 1)$ such that $a_\lambda := \lambda \tilde{a} + (1 - \lambda)\hat{a}$ is in $F(J)$.

Hence, $c_0(\bar{a}) < c_0(\hat{a})$ would imply $c_0(a_\lambda) < c_0(\hat{a})$ and so contradict the optimality of \hat{a} on $F(J)$. \square

We introduce now the subsequent notion of convexity.

Definition 4.2 (cf. Krabs [26]). A convex function c on a convex set $C \subseteq \mathbb{R}^n$ is said to be (almost) strictly convex if

$$a, b \in C, a \neq b, \lambda \in (0, 1), (c(a) = c(b)) \Rightarrow c(\lambda a + (1 - \lambda)b) < \lambda c(a) + (1 - \lambda)c(b).$$

An almost strictly convex function can have at most one minimal point. In particular, each strictly convex function is almost strictly convex, and, for example, the function $c(a) := \|a\|_2$, $a \in \mathbb{R}^n$, has this property. We supplement next the assumptions of Section 0 by the following ones.

(A3) $A \subseteq \mathbb{R}^n$ is convex, g is convex and f is almost strictly convex on an open set $A_0 \supseteq A$. Further, f_k , $k \in \mathbb{N}_0$, is an arbitrarily chosen sequence of convex (and hence continuous) functions on A_0 with (1.5) and

$$f_k(a) \leq f_{k+1}(a) \leq f(a), \quad k \in \mathbb{N}_0, a \in A_0. \quad (4.1)$$

Under assumption (A3) the inactive constraints in Algorithms I and II can be dropped after the solution of each subproblem. Thus we can state the succeeding modifications of these algorithms.

Algorithm III. Let (A3) and (A0) with convex sets X and \tilde{X} be satisfied. Let further B_k , $k \in \mathbb{N}_0$, be as in (2.5) and ε_k, δ_k , $k \in \mathbb{N}_0$, as in (3.1).

Step 0. Set $M_0 := X$, $\tilde{M}_0 := \tilde{X}$, $J_0 := \emptyset$, $s_j(a) := -\infty$, $a \in A$, $j \in J_0$, and $k := 0$.

Step 1. For

$$(P_k): \quad \rho_k := \inf_{a \in M_k} f_k(a),$$

find an (approximate) solution such that $\tilde{a}_k \in \tilde{M}_k$ and

$$\|\tilde{a}_k - a_k\| \leq \varepsilon_k, \quad (4.2)$$

where $a_k \in M_k$ solves (P_k) , i.e., satisfies $f_k(a_k) = \rho_k$.

Step 2. Determine $E_k \subseteq J_k$ with $E_k \supseteq \{j \in J_k \mid s_j(a_k) = 0\}$ and $D_{k+1} \subseteq B_{k+1}$ finite with $D_{k+1} \supseteq \{x_k\}$ where

$$g(\tilde{a}_k, x_k) = \max_{x \in B_{k+1}} g(\tilde{a}_k, x).$$

For $\delta \geq 0$, define

$$N_k(\delta) := \{a \in A \mid s_j(a) \leq \delta, j \in E_k; g(a, x) \leq \delta, x \in D_{k+1}\}. \quad (4.3)$$

Rename the constraints in $N_k(\delta)$ such that, with some index set J_{k+1} ,

$$N_k(\delta) = \{a \in A \mid s_j(a) \leq \delta, j \in J_{k+1}\},$$

and set

$$M_{k+1} := X \cap N_k(0), \quad \tilde{M}_{k+1} := \tilde{X} \cap N_k(\delta_{k+1}). \quad (4.4)$$

Step 3. Set $k := k + 1$ and go to Step 1.

Remark 4.3. By means of $E(a_k) := \{x \in D_k \mid g(a_k, x) = 0\}$ and $D_0 = \emptyset$, the set (4.3) may alternatively be defined by choosing $D_{k+1} \subseteq B_{k+1}$ finite with $D_{k+1} \supseteq E(a_k) \cup \{x_k\}$ and setting

$$N_k(\delta) := \{a \in A \mid g(a, x) \leq \delta, x \in D_{k+1}\} = Z_\delta(D_{k+1}). \quad (4.5)$$

Further, the set B_{k+1} in the algorithm may be exchanged for B_k as well.

The corresponding modification of Algorithm II is the following.

Algorithm IV. As Algorithm III with (4.3) exchanged for

$$N_k(\delta) := \left\{a \in A \mid s_j(a) \leq \delta, j \in E_k; g(\tilde{a}_k, x) + \partial g(\tilde{a}_k, x)^T(a - \tilde{a}_k) \leq \delta, x \in D_{k+1}\right\}. \quad (4.6)$$

Theorem 4.4. Algorithm III or Algorithm IV, respectively, generate a sequence $\tilde{a}_k \in \tilde{X}$, $k \in \mathbb{N}_0$, which is minimizing for $S[B]$. Moreover, $\tilde{a}_k \rightarrow \tilde{a}$, $k \in \mathbb{N}_0$, holds true, where $\tilde{a} \in Z_0(B)$ is the unique solution of $S[B]$, and

$$\rho_k \leq \rho_{k+1} \leq \mu(B), \quad k \in \mathbb{N}_0. \quad (4.7)$$

Proof. By virtue of our assumptions, (P_k) possesses a solution for each $k \in \mathbb{N}_0$, and $S[B]$ has a unique solution \tilde{a} . In Step 1, therefore, an element \tilde{a}_k satisfying (4.2) exists. As in the proof of Theorem 3.1 we now let $M := L(a_0, B)$ be the feasible set of $S[B]$. Then from $Z_0(B) \subseteq N_k(0)$ we infer $M \subseteq M_k$, $k \in \mathbb{N}_0$, so that (1.1)–(1.4) are satisfied. The convergence (1.7) can be deduced from (4.2), (3.1) and the compactness of \tilde{X} . So, if we verify (1.6), we can appeal to Theorem 1.2.

Let $\mathbb{N}' \subseteq \mathbb{N}_0$ and $\tilde{a}_k \rightarrow a^* \in \tilde{X}$, $k \in \mathbb{N}'$. Then there exist $\mathbb{N}'' \subseteq \mathbb{N}'$ and $a^{**} \in \tilde{X}$ such that $\tilde{a}_{k+1} \rightarrow a^{**}$, $k \in \mathbb{N}''$. Thus by (4.2) and (3.1) we have

$$a_k \rightarrow a^*, \quad k \in \mathbb{N}'', \quad a_{k+1} \rightarrow a^{**}, \quad k \in \mathbb{N}''.$$

Our convexity assumptions enable us to apply Lemma 4.1 so that because of $s_j(a_{k+1}) \leq 0$, $j \in E_k$, $k \in \mathbb{N}_0$, and the optimality of a_k for (P_k) we arrive at

$$f_k(a_k) \leq f_k(\lambda a_k + (1 - \lambda)a_{k+1}), \quad \lambda \in [0, 1], \quad k \in \mathbb{N}_0, \quad (4.8)$$

and

$$f_k(a_k) \leq f_k(a_{k+1}) \leq f_{k+1}(a_{k+1}) \leq \mu(B), \quad k \in \mathbb{N}_0, \quad (4.9)$$

where for (4.9) also condition (4.1) has been used. Eq. (4.9) in particular proves (4.7). Utilizing (1.5), from (4.8) and (4.9) we further derive

$$f(a^*) \leq f(\lambda a^* + (1 - \lambda)a^{**}), \quad \lambda \in [0, 1], \quad f(a^*) = f(a^{**}).$$

Since f is almost strictly convex on A , this implies $a^* = a^{**}$. Clearly, a^* is in A and, because of $f_k(a_k) \leq f(a_0)$ with a_0 from (A0), a^* also satisfies $f(a^*) \leq f(a_0)$. So it remains to show $g(a^*, x) \leq 0$, $x \in B$.

For $\xi \in B$ we can find $\xi_k \in B_{k+1}$, $k \in \mathbb{N}''$, such that $\xi_k \rightarrow \xi$, $k \in \mathbb{N}''$. With some $\mathbb{N}''' \subseteq \mathbb{N}''$ and $x^* \in B$ we have further

$$\tilde{a}_k \rightarrow a^*, \quad k \in \mathbb{N}''', \quad x_k \rightarrow x^*, \quad k \in \mathbb{N}'''.$$

So, because of $g(\tilde{a}_k, x_k) \geq g(\tilde{a}_k, \xi_k)$, $k \in \mathbb{N}'''$, we get

$$g(a^*, x^*) \geq g(a^*, \xi).$$

For Algorithm III we know now that $g(\tilde{a}_{k+1}, x_k) \leq \delta_{k+1}$, $k \in \mathbb{N}'''$, whereas in case of Algorithm IV we have

$$g(\tilde{a}_k, x_k) + \partial g(\tilde{a}_k, x_k)^T (\tilde{a}_{k+1} - \tilde{a}_k) \leq \delta_{k+1}, \quad k \in \mathbb{N}'''.$$

By taking the limit, we arrive in both cases at $g(a^*, x^*) \leq 0$ so that the proof is complete. \square

Remark 4.5. Remark 3.3 likewise applies to Algorithms III and IV. With regard to (4.6), we also refer to Remark 3.7. When an exact solution a_k of (P_k) in Step 1 of both algorithms is not known, a suitable choice of E_k in Step 2 is

$$E_k := \{j \in J_k \mid s_j(\tilde{a}_k) \geq -\sigma_k\}, \quad (4.10)$$

with some proper $\sigma_k > 0$. (Due to (4.2) and the continuity of s_j , $j \in J_k$, on \tilde{X} , then for suitable σ_k the set E_k is of the requested type.) The efficiency of Algorithms III and IV heavily depends on the choice of the sets D_k and B_k , $k \in \mathbb{N}$. Our numerical results in [32,33] suggest to specify D_{k+1} by

$$D_{k+1} := \{x \in B_{k+1} \mid g(\tilde{a}_k, x) \geq \eta_{k+1}\}, \quad k \in \mathbb{N}_0,$$

with some reasonable $\eta_{k+1} \in \mathbb{R}$, $k \in \mathbb{N}_0$, and to choose $B_{k+1} = B_k$, until nearly $g(\tilde{a}_k, x_k) \leq 0$ is obtained and hence \tilde{a}_k solves $S[B_k]$ approximately. The latter ideas will be used in Algorithm V.

Remark 4.6. For the finite case, where $B_k := B$, $|B_k| < \infty$, $f_k := f$, $\tilde{a}_k := a_k$, $k \in \mathbb{N}$, Algorithm IV has essentially been given in [5,37,38]. (In our opinion strict quasi-concavity, suggesting strict quasi-convexity of f here, does not suffice to guarantee [5, (2)] and [38, (4)], respectively. Consider, for example, the strictly quasi-concave function c on \mathbb{R}^2 with $c(a_1, a_2) = 0$ for $a_1 \leq 0$ and $c(a_1, a_2) = -a_1$ for $a_1 \geq 0$.) For the semi-infinite case ($B_k := B$, $f_k := f$ being strictly convex, $\tilde{a}_k := a_k$, $D_{k+1} := \{x_k\}$, $k \in \mathbb{N}$, and $\tilde{X} := X$ not being specified), convergence of Algorithm III has been proved in [1] whereas Algorithm IV (with $B_k \neq B$, $k \in \mathbb{N}$, being permitted) has been suggested in [36]. (The latter paper is partially inaccurate; in particular the condition

$T_m \subseteq T_{m+1}$, $m \in \mathbb{N}$, has to be added there.)

We remark that the authors of [9] consider constraint dropping schemes for algorithms searching for stationary points of semi-infinite programming problems. It might be interesting to investigate whether the frame of Section 1 can be modified in order to include also such methods. For linear semi-infinite programming problems, moreover, in [2] (under the restrictive Haar condition) and in [22] exchange methods of the above type are presented where at each iteration exactly n constraints are employed. As Theorem 4.4 indicates, it will generally be difficult to prove convergence of such “explicit exchange methods” without assuming some kind of strict convexity (like Haar’s condition), for which reason these methods have not been studied much. (The exchange rules in [22] are rather complicated, and no numerical experience for the algorithms has been reported there.) We finally mention that also the methods in [15,16] and [32, Algorithm 3.1] are related to Algorithms III and IV. We discuss these latter methods next in the following section.

5. An algorithm for general convex semi-infinite optimization problems

For the algorithms of the preceding section it seems to be possible to prove convergence only when the objective function of the considered problem is almost strictly convex. Therefore, these algorithms are especially not applicable to linear problems. So at first sight it is confusing that there exist very similar-looking methods for the solution of linear semi-infinite programming problems in [15,32]. (The algorithm of [15] has also been implemented for strictly convex quadratic programs with linear constraints in [16].) These methods, however, are distinguished from Algorithms III and IV, as they do not make explicit use of a compact set X enclosing the region of feasible points and as their convergence is based on the fact that discrete problems $S[B_i]$, $i = 0, 1, \dots$, are solved (almost) completely. This in particular implies that (under a proper compactness assumption) the minimizing property for $S[B]$ can only be verified for the subsequence \tilde{a}_i , $i \in \mathbb{N}_0$, of iterates solving $S[B_i]$, $i \in \mathbb{N}_0$. (Under (A1) this result follows for the algorithms in [15,16,32] from Theorem 3.1 with Remark 3.2 here, if $D_k := B_k$ and $f_k := f$, $k \in \mathbb{N}_0$, is chosen in Algorithm I.)

Remark 5.1. If the feasible sets of the subproblems in the algorithm of [16] are intersected with a compact set X as provided by (A1) here, this algorithm becomes a special implementation of Algorithm III or IV, respectively. Moreover by, Theorem 4.4, then the total sequence of points generated by the algorithm converges to the solution of the corresponding problem. Since for linear problems assumption (A3) is not satisfied, the same arguments do not apply to the algorithms in [15,32].

For the algorithms in [15,16,32] it has been shown that they are able to solve quite large linear and linearly constrained quadratic semi-infinite programming problems. Employing the results of the previous sections, we want to provide here a similar algorithm for general convex semi-infinite programs $S[B]$ in which (for a specification) quadratic subproblems have to be solved.

As in [15,16,32] we start from a sequence of grids B_i , $i \in \mathbb{N}_0$, satisfying (2.5) where it is reasonable (but not necessary) to let $B_{i+1} \setminus B_i \neq \emptyset$, $i \in \mathbb{N}_0$. Next we add the following supposition to our general assumptions of Section 0.

(A4) $A \subseteq \mathbb{R}^n$ is convex, g and f are convex on an open set $A_0 \supseteq A$, and f_i , $i \in \mathbb{N}_0$, is an arbitrarily given sequence of almost strictly convex (and hence continuous) functions on A_0 fulfilling (1.5).

Since each convex optimization problem can be equivalently transformed into one with a linear objective function, the function f may assumed to be linear, if requested, and thus the f_i , $i \in \mathbb{N}_0$, may be simple quadratic functions chosen by the user. Assuming (A4) and (A0) or (A1), we note now that by virtue of Theorem 3.1 (with $D_k := B_k$, $k \in \mathbb{N}_0$, in Algorithm I) any sequence of approximate and increasingly more accurate solutions of the problems

$$\tilde{\rho}_i := \inf_{a \in (X \cap Z_0(B_i))} f_i(a) \quad \text{or} \quad \tilde{\rho}_i := \inf_{a \in Z_0(B_i)} f_i(a), \quad i \in \mathbb{N}_0, \quad (5.1)$$

respectively, is a minimizing sequence for $S[B]$ (cf. Remark 3.2), and that by Theorem 4.4, at least for quadratic f_i and polyhedral X , an approximate solution of (5.1) can be evaluated through finitely many iterations of Algorithm IV, if each subproblem (P_k) there is equipped with the goal function f_i and solved exactly. These observations lead to the following algorithm where, in order to make the algorithm efficient, we have further to guarantee that the constraint dropping strategy of Algorithm IV for solving (5.1) can be continued in that iteration in which the grid index is raised by one.

In the algorithm, the first index i refers to the current grid B_i and to the problem (5.1) currently being solved, whereas the second index counts the iterations of the inner algorithm for the solution of this problem. For a first reading it may be helpful to set $\tilde{X} := X$ and to choose all numbers in (5.2) and (5.3) identically zero such that in particular $\tilde{M}_{i,k} = M_{i,k}$.

Algorithm V. Let (A4) and (A0) with convex sets X , \tilde{X} be satisfied. Moreover, let B_i , $i \in \mathbb{N}_0$, be as in (2.5), and let $\varepsilon_{i,k}$, $\delta_{i,k}$, ζ_i , ϑ_i be given numbers such that

$$\min(\zeta_i, \vartheta_i) \geq 0, \quad i \in \mathbb{N}_0, \quad \max(\zeta_i, \vartheta_i) \rightarrow 0, \quad i \in \mathbb{N}_0, \quad (5.2)$$

and, for each $i \in \mathbb{N}_0$,

$$\min(\varepsilon_{i,k}, \delta_{i,k}) \geq 0, \quad k \in \mathbb{N}_0, \quad \max(\varepsilon_{i,k}, \delta_{i,k}) \rightarrow 0, \quad k \in \mathbb{N}_0. \quad (5.3)$$

Step 0. Set $M_{0,0} := X$, $\tilde{M}_{0,0} := \tilde{X}$, $J_{0,0} := \emptyset$, $s_j(a) := -\infty$, $a \in A$, $j \in J_{0,0}$, $k := 0$, and $i := 0$.

Step 1. For

$$(P_{i,k}): \quad \rho_{i,k} := \inf_{a \in M_{i,k}} f_i(a),$$

find an (approximate) solution such that $\tilde{a}_{i,k} \in \tilde{M}_{i,k}$ and

$$\|\tilde{a}_{i,k} - a_{i,k}\| \leq \varepsilon_{i,k}, \quad (5.4)$$

where $a_{i,k} \in M_{i,k}$ solves $(P_{i,k})$, i.e., satisfies $f_i(a_{i,k}) = \rho_{i,k}$.

Step 2. Compute $x_{i,k} \in B_i$ with

$$g(\tilde{a}_{i,k}, x_{i,k}) = \max_{x \in B_i} g(\tilde{a}_{i,k}, x).$$

If

$$g(\tilde{a}_{i,k}, x_{i,k}) \leq \zeta_i, \quad \left| f_i(\tilde{a}_{i,k}) - \inf_{a \in (X \cap Z_0(B_i))} f_i(a) \right| \leq \vartheta_i, \quad (5.5)$$

go to Step 3, where the second criterion in (5.5) can be disregarded for $\tilde{a}_{i,k} = a_{i,k}$ (see Remark 5.4). Otherwise choose $E_{i,k} \subseteq J_{i,k}$ with $E_{i,k} \supseteq \{j \in J_{i,k} \mid s_j(a_{i,k}) = 0\}$ and $D_{i,k+1} \subseteq B_i$ finite with $D_{i,k+1} \supseteq \{x_{i,k}\}$ and let

$$N_{i,k}(\delta) := \left\{ a \in A \mid s_j(a) \leq \delta, j \in E_{i,k}; g(\tilde{a}_{i,k}, x) + \partial g(\tilde{a}_{i,k}, x)^T(a - \tilde{a}_{i,k}) \leq \delta, x \in D_{i,k+1} \right\},$$

for $\delta \geq 0$. Rename the constraints in $N_{i,k}(\delta)$ such that, with some index set $J_{i,k+1}$,

$$N_{i,k}(\delta) = \{a \in A \mid s_j(a) \leq \delta, j \in J_{i,k+1}\},$$

and define

$$M_{i,k+1} := X \cap N_{i,k}(0), \quad \tilde{M}_{i,k+1} := \tilde{X} \cap N_{i,k}(\delta_{i,k+1}).$$

Set $k := k + 1$ and go to Step 1.

Step 3. Set $k_i^* := k$, $a_i^* := \tilde{a}_{i,k}$, $M_{i+1,0} := M_{i,k}$, $\tilde{M}_{i+1,0} := \tilde{M}_{i,k}$, $J_{i+1,0} := J_{i,k}$. Now let $i := i + 1$ and $k := 0$, and go to Step 1.

Remark 5.2. $\zeta_i = \vartheta_i = 0$ and $\varepsilon_{i,k} = \delta_{i,k} = 0$ in (5.2) and (5.3) are only permitted if problem (5.1) or $(P_{i,k})$, respectively, is solvable by means of finitely many operations. For $(P_{i,k})$ this is possible when f_i also is quadratic and X is a polyhedron, and for (5.1), if in addition g is a linear map. (The latter follows from the finiteness of B_i and the facts that $a_{i,k+1} \neq a_{i,k}$ for i fixed and hence $f_i(a_{i,k}) < f_i(a_{i,k+1})$ is true.)

Theorem 5.3. The sequence a_i^* , $i \in \mathbb{N}_0$, generated by Algorithm V, is a minimizing sequence for $S[B]$.

Proof. We first note that, given $m \in \mathbb{N}_0$, an index set J_m and functions

$$s_j(a) := g(b_j, x) + \partial g(b_j, x)^T(a - b_j), \quad a \in A, j \in J_m,$$

for some $b_j \in \mathbb{R}^n$ and $x \in B_m$ defining

$$N_{m-1}(\delta) := \{a \in A \mid s_j(a) \leq \delta, j \in J_m\},$$

Theorem 4.4 remains true for $k \geq m$ with regard to Algorithm IV if Step 0 there is exchanged for the following.

Step 0. Set $M_m := X \cap N_{m-1}(0)$, $\tilde{M}_m := \tilde{X} \cap N_{m-1}(\delta_m)$, and $k := m$.

We let now $i \in \mathbb{N}_0$ be fixed and consider the finite convex optimization problem

$$(\tilde{P}_i): \quad \tilde{\rho}_i := \inf_{a \in (X \cap Z_0(B_i))} f_i(a), \quad (5.6)$$

having an almost strictly convex goal function f_i . (When we apply results of previous sections to (\tilde{P}_i) , we set $B_k := B$, $k \in \mathbb{N}_0$, and $f_k := f$, $k \in \mathbb{N}_0$, there and then let $B := B_i$ and $f := f_i$ with B_i, f_i from (\tilde{P}_i) .) Assumption (A0), providing compact sets X, \tilde{X} , implies that a_0 is in $X \cap Z_0(B_i)$, that the set $\hat{L}_i := \{a \in X \mid f_i(a) \leq f_i(a_0)\} \cap Z_0(B_i)$ is compact and that $\hat{L}_i \subseteq X \subseteq \tilde{X}$, i.e., that (A0) is also fulfilled for (\tilde{P}_i) with the same sets X, \tilde{X} . Thus, according to Theorem 4.4, Algorithm IV can be employed for the solution of (\tilde{P}_i) . By our comment at the beginning of this proof, it can be started with the sets $M_{i,0}, \tilde{M}_{i,0}$ and $J_{i,0}$. (Note that $B_l \subseteq B_i$ for all $l \in \mathbb{N}_0$ with $0 \leq l \leq i$.) Consequently, for $\zeta_i = \vartheta_i = 0$, the sequence $\tilde{a}_{i,k}$, $k \in \mathbb{N}_0$, converges to the unique solution of (\tilde{P}_i) . This shows that the index i in Algorithm V is increased by one after finitely many iterations (cf. the proof of Theorem 4.4 and Remark 5.2).

Due to (5.5), we have $a_i^* \in Z_{\zeta_i}(B_i)$ and $|f_i(a_i^*) - \tilde{\rho}_i| \leq \vartheta_i$ for $i \in \mathbb{N}_0$. By setting $D_k := B_k$, $k \in \mathbb{N}_0$, in Algorithm I, we can finally conclude from Theorem 3.1 that the sequence a_i^* , $i \in \mathbb{N}_0$, is minimizing for $S[B]$. \square

Remark 5.4. If assumption (A0) in the algorithm is replaced by (A1), then one can employ the identity

$$\tilde{\rho}_i = \inf_{a \in (X \cap Z_0(B_i))} f_i(a) = \inf_{a \in Z_0(B_i)} f_i(a), \quad i \in \mathbb{N}_0,$$

in (5.5) (see (3.11)). By Theorem 4.4 the minimal values of $(P_{i,k})$ satisfy the relation

$$\rho_{i,k} \leq \rho_{i,k+1} \leq \tilde{\rho}_i, \quad k = 0, \dots, k_i^* - 1, \quad i \in \mathbb{N}_0,$$

where $\tilde{\rho}_i \rightarrow \mu(B)$, $i \in \mathbb{N}_0$. In case

$$f_i(a) \leq f_{i+1}(a) \leq f(a), \quad a \in A, \quad i \in \mathbb{N}_0, \quad (5.7)$$

we can further obtain

$$\rho_{i,0} \leq \dots \leq \rho_{i,k_i^*} \leq \dots \leq \rho_{i+1,k_{i+1}^*} \leq \mu(B), \quad i \in \mathbb{N}_0,$$

which because of $|\rho_{i,k_i^*} - \tilde{\rho}_i| \rightarrow 0$, $i \in \mathbb{N}_0$, (by (5.4), and (5.5)) implies the monotonic convergence of $\rho_{i,k}$, $k = 0, \dots, k_i^*$, $i \in \mathbb{N}_0$, to $\mu(B)$. Concerning an explicit choice of the sets B_i , $E_{i,k}$, and $D_{i,k}$ in Algorithm V, we refer to Remark 4.5 and to the numerical examples in [32,33]. Finally, it is remarkable that the second and usually not verifiable criterion in (5.5) can be erased if the exact solution $\tilde{a}_{i,k} := a_{i,k}$ of $(P_{i,k})$ can be computed. (Let $U_i(\zeta) := X \cap Z_{\zeta}(B_i)$ and

$$d_i(\zeta) := \max_{a \in U_i(\zeta)} \min_{b \in U_i(0)} \|a - b\|,$$

and define

$$\vartheta_i := \sup\{|f_i(a) - f_i(b)| \mid \|a - b\| \leq d_i(\zeta_i), a \in X, b \in X\};$$

then for $a_{i,k}$ satisfying the first inequality in (5.5), i.e., for $a_{i,k} \in U_i(\zeta_i)$, we have

$$0 \leq \tilde{\rho}_i - \rho_{i,k} = \inf_{a \in U_i(0)} f_i(a) - f_i(a_{i,k}) \leq \vartheta_i,$$

and for X and $\zeta_i, f_i, i \in \mathbb{N}_0$, as required in the algorithm, it can be shown that $d_i(\zeta_i) \rightarrow 0, i \rightarrow \infty$, and hence $\vartheta_i \rightarrow 0, i \rightarrow \infty$.)

Remark 5.5. Let $0 \leq \gamma_{i+1} \leq \gamma_i, i \in \mathbb{N}_0$, and $\gamma_i \rightarrow 0, i \in \mathbb{N}_0$. Possible choices for $f_i, i \in \mathbb{N}_0$, in Algorithm V are the following:

$$f_i(a) := f(a) + \gamma_i \|a\|_2^2, \quad a \in \tilde{X}, \quad i \in \mathbb{N}_0, \quad (5.8)$$

$$f_i(a) := f(a) + \gamma_i \|a - a_{i-1}^*\|_2^2, \quad a \in \tilde{X}, \quad i \in \mathbb{N},$$

$$f_i(a) := f(a) + \gamma_i (\|a\|_2^2 - C), \quad a \in \tilde{X}, \quad i \in \mathbb{N}_0, \quad \text{with } C \geq \max\{\|a\|_2^2 \mid a \in \tilde{X}\}. \quad (5.9)$$

If \tilde{X} is a polyhedron, $\max\{\|a\|_2^2 \mid a \in \tilde{X}\}$ can be determined numerically (cf. [20]). The functions (5.9) in particular satisfy (5.7).

If X is a polyhedron and if each problem $(P_{i,k})$ is solved exactly (which is possible if the $f_i, i \in \mathbb{N}_0$, are chosen to be strictly convex quadratic functions), each step in Algorithm V can be performed by means of finitely many operations. In particular, this implies that under (A0) each convex semi-infinite programming problem can be solved by the iterative solution of a sequence of quadratic problems with a strictly convex goal function and linear constraints.

Algorithm V has been specified and implemented for the solution of minimax problems in the complex domain (see [33]). The results in [33] show that the algorithm can yield highly accurate solutions also for larger and difficult problems. (Six to ten significant digits of solutions of problems with up to 61 unknowns are provided there.) The algorithm of [33] was further successfully used in [30] to design various types of digital filters with up to 800 unknowns. We remark that for the problems in [30,33], the idea of working on a sequence of finite grids here can be efficiently combined with the Fast Fourier Transformation in order to compute the maximum value in Step 2 of Algorithm V.

Acknowledgement

The author is grateful to an anonymous referee for valuable comments which have helped to improve the presentation of the results.

References

- [1] J.W. Blankenship and J.E. Falk, Infinitely constrained optimization problems, *J. Optim. Theory Appl.* **19** (1976) 261–281.
- [2] E. Blum and W. Oettli, *Mathematische Optimierung* (Springer, Berlin, 1975).
- [3] E.W. Cheney and A.A. Goldstein, Newton's method for convex programming and Tchebycheff approximation, *Numer. Math.* **1** (1959) 253–268.
- [4] J.J. Dinkel, W.H. Elliott and G.A. Kochenberger, Computational aspects of cutting-plane algorithms for geometric programming problems, *Math. Programming* **13** (1977) 200–220.
- [5] B.C. Eaves and W.I. Zangwill, Generalized cutting plane algorithms, *SIAM J. Control* **9** (1971) 529–542.

- [6] J. Elzinga and T.G. Moore, A central cutting plane algorithm for the convex programming problem, *Math. Programming* **8** (1975) 134–145.
- [7] M. Fukushima, An outer approximation algorithm for solving general convex programs, *Oper. Res.* **31** (1983) 101–113.
- [8] M. Fukushima, On the convergence of a class of outer approximation algorithms for convex programs, *J. Comput. Appl. Math.* **10** (2) (1984) 147–156.
- [9] C. Gonzaga and E. Polak, On constraint dropping schemes and optimality functions for a class of outer approximations algorithms, *SIAM J. Control Optim.* **17** (1979) 477–493.
- [10] P.R. Gribik, A central-cutting-plane algorithm for semi-infinite programming problems, in: R. Hettich, Ed., *Semi-Infinite Programming* (Springer, Berlin, 1979) 66–82.
- [11] R.D. Grigorieff and R.M. Reemtsen, Discrete approximations of minimization problems. I. Theory, *Numer. Funct. Anal. Optim.* **11** (1990) 701–719.
- [12] S.-Å. Gustafson and K.O. Kortanek, Numerical treatment of a class of semi-infinite programming problems, *Naval Res. Logist. Quart.* **20** (1973) 477–504.
- [13] S.-Å. Gustafson and K.O. Kortanek, Semi-infinite programming and applications, in: A. Bachem, M. Grötschel and B. Korte, Eds., *Mathematical Programming. The State of the Art* (Springer, Berlin, 1982) 132–157.
- [14] R. Hettich, A review of numerical methods for semi-infinite optimization, in: A.V. Fiacco and K.O. Kortanek, Eds., *Semi-Infinite Programming and Applications* (Springer, Berlin, 1983) 158–178.
- [15] R. Hettich, An implementation of a discretization method for semi-infinite programming, *Math. Programming* **34** (1986) 354–361.
- [16] R. Hettich and G. Gramlich, A note on an implementation of a method for quadratic semi-infinite programming, *Math. Programming* **46** (1990) 249–254.
- [17] R. Hettich and P. Zencke, *Numerische Methoden der Approximation und Semi-Infiniten Optimierung* (Teubner, Stuttgart, 1982).
- [18] W.W. Hogan, Applications of a general convergence theory for outer approximation algorithms, *Math. Programming* **5** (1973) 151–168.
- [19] R. Horst, Deterministic global optimization: some recent advances and new fields of application, *Naval Res. Logist. Quart.* **37** (1990) 433–471.
- [20] R. Horst and H. Tuy, *Global Optimization* (Springer, Berlin, 1990).
- [21] H. Hu, A one-phase algorithm for semi-infinite linear programming, *Math. Programming* **46** (1990) 85–103.
- [22] U. Juergens, Zur Konvergenz semiinfiniter Mehrfachtauschalgorithmen, Ph.D. Thesis, Univ. Hamburg, 1986.
- [23] K.C. Kapur, On cutoff optimization methods in infinite-dimensional spaces and applications, *J. Optim. Theory Appl.* **12** (1973) 16–31.
- [24] J.E. Kelley Jr, The cutting-plane method for solving convex programs, *J. SIAM* **8** (1960) 703–712.
- [25] P. Kosmol, *Methoden zur Numerischen Behandlung Nichtlinearer Gleichungen und Optimierungsaufgaben* (Teubner, Stuttgart, 1989).
- [26] W. Krabs, *Einführung in die Kontrolltheorie* (Wissenschaftliche Buchgesellschaft, Darmstadt, 1978).
- [27] E.S. Levitin and B.T. Polyak, Constrained optimization methods, *U.S.S.R. Comput. Math. and Math. Phys.* **6** (1966) 1–50.
- [28] E.R. Panier and A.L. Tits, A globally convergent algorithm with adaptively refined discretization for semi-infinite optimization problems arising in engineering design, *IEEE Trans. Automat. Control.* **AC-34** (1989) 903–908.
- [29] S.C. Parikh, Approximate cutting planes in nonlinear programming, *Math. Programming* **11** (1976) 194–198.
- [30] A. Potchinkov and R. Reemtsen, FIR filter design in the complex domain by a semi-infinite programming technique (I. The method), *Arch. Elektronik Übertragungstechnik* **48** (1994) 135–144; (II. Numerical results), to appear.
- [31] R. Reemtsen, Modifications of the first Remez algorithm, *SIAM J. Numer. Anal.* **27** (1990) 507–518.
- [32] R. Reemtsen, Discretization methods for the solution of semi-infinite programming problems, *J. Optim. Theory Appl.* **71** (1991) 85–103.
- [33] R. Reemtsen, A cutting plane method for solving minimax problems in the complex plane, *Numer. Algorithms* **2** (1992) 409–436.

- [34] Y. Tanaka, M. Fukushima and T. Ibaraki, A globally convergent SQP method for semi-infinite nonlinear optimization, *J. Comput. Appl. Math.* **23** (2) (1988) 141–153.
- [35] R. Tichatschke and T. Lohse, Eine verallgemeinerte Schnittmethode für konvexe semi-infinite Optimierungsaufgaben, *Wiss. Z. Tech. Hochsch. Karl-Marx-Stadt* **24** (1982) 332–338.
- [36] R. Tichatschke and V. Nebeling, A cutting-plane method for quadratic semi-infinite programming problems, *Optimization* **19** (1988) 803–817.
- [37] D.M. Topkis, Cutting-plane methods without nested constraint sets, *Oper. Res.* **18** (1970) 404–413.
- [38] D.M. Topkis, A note on cutting-plane methods without nested constraint sets, *Oper. Res.* **18** (1970) 1216–1220.
- [39] D.M. Topkis, A cutting-plane algorithm with linear and geometric rates of convergence, *J. Optim. Theory Appl.* **36** (1982) 1–22.
- [40] A.F. Veinott Jr, The supporting hyperplane method for unimodal programming, *Oper. Res.* **15** (1967) 147–152.